



«Серия "Недостающее руководство" — просто самая разумная и полезная серия руководств».
КЕВИН КЕЛЛИ, СОУЧРЕДИТЕЛЬ ЖУРНАЛА «WIRED»

HTML5

недостающее руководство

Книга, которая должна быть на полке



O'REILLY®

Мэтью Мак-Дональд

HTML5

the missing manual[®]

The book that should have been in the box[®]

Matthew MacDonald

O'REILLY[®]

Beijing | Cambridge | Farnham | Köln | Sebastopol | Tokyo

Мэтью Мак-Дональд

HTML5

недостающее руководство

Санкт-Петербург

«БХВ-Петербург»

2012

УДК 681.3.068
ББК 32.973.26-018.1
М15

Мак-Дональд М.

М15 HTML5. Недостающее руководство: Пер. с англ. — СПб.: БХВ-Петербург, 2012. — 480 с.: ил.

ISBN 978-5-9775-0805-6

Доступно и в занимательной форме рассказано, как HTML превратился в HTML5. Рассмотрены семантические элементы и новые стандарты языка. Описано, как создавать современные веб-страницы, в том числе улучшенные веб-формы, поддерживать аудио и видео, рисовать на холсте, совершенствовать оформление веб-страниц с помощью CSS3. Даны практические рекомендации по созданию интеллектуальных веб-приложений, хранению данных, разработке автономных приложений. Показано, как реализовать взаимодействие с веб-сервером, геолокацию, фоновые вычисления, управление историей просмотров и браузерную совместимость с элементами HTML5.

Для веб-разработчиков

УДК 681.3.068
ББК 32.973.26-018.1

Группа подготовки издания:

Главный редактор	<i>Екатерина Кондукова</i>
Зам. главного редактора	<i>Евгений Рыбаков</i>
Зав. редакцией	<i>Елена Васильева</i>
Перевод с английского	<i>Сергея Таранушенко</i>
Редактор	<i>Анна Кузьмина</i>
Компьютерная верстка	<i>Ольги Сергиенко</i>
Корректор	<i>Зинаида Дмитриева</i>
Оформление обложки	<i>Марины Дамбиевой</i>

Authorized translation of the English edition of HTML5: The Missing Manual by Matthew MacDonald, ISBN: 978-1-449-30239-9, Copyright © 2011 O'Reilly Media, Inc. This translation is published and sold by permission of O'Reilly Media, Inc., the owner of all rights to publish and sell the same.

Авторизованный перевод английской редакции книги: ISBN: 978-1-449-30239-9, Copyright © 2011 O'Reilly Media, Inc. Перевод опубликован и продается с разрешения O'Reilly Media, Inc., собственника всех прав на публикацию и продажу издания.

Подписано в печать 30.04.12.
Формат 70×100^{1/16}. Печать офсетная. Усл. печ. л. 38,7.
Тираж 2000 экз. Заказ №
"БХВ-Петербург", 190005, Санкт-Петербург, Измайловский пр., 29.
Первая Академическая типография "Наука"
199034, Санкт-Петербург, 9 линия, 12/28

ISBN 978-1-449-30239-9 (англ.)
ISBN 978-5-9775-0805-6 (рус.)

© 2011 O'Reilly Media, Inc.
© Перевод на русский язык "БХВ-Петербург", 2012

Оглавление

Об авторе.....	13
Благодарности.....	13
Введение	15
Что нужно знать для работы с этой книгой?	16
Написание кода HTML5	17
Просмотр страницы HTML5.....	18
Когда HTML5 будет готов?	18
О чем эта книга?	19
Онлайновые ресурсы	21
Недостающий CD.....	21
Сайт для тестирования примеров книги	21
ЧАСТЬ I. ЗНАКОМСТВО С НОВЫМ ЯЗЫКОМ.....	23
Глава 1. Представляем HTML5.....	25
История языка HTML5.....	25
Язык XHTML 1.0: строго по правилам.....	26
XHTML 2: неожиданный провал.....	27
HTML5: возвращение к жизни	27
HTML: живой язык	29
Три основных принципа HTML5	30
Принцип 1. Не рвите Паутину	30
Принцип 2. Асфальтируйте тропинки.....	32
Принцип 3. Будьте практичными	32
Первое знакомство с разметкой HTML5	33
Описание типа документа HTML5	35
Кодировка символов.....	37
Язык	37
Добавление таблицы стилей	38
Добавление JavaScript-кода.....	38
Конечный результат.....	39

Углубленное знакомство с синтаксисом HTML5.....	40
Ослабленные правила	40
Проверка кода HTML5	41
Возвращение XHTML.....	44
Семейство элементов HTML5	46
Добавленные элементы	46
Удаленные элементы	47
Адаптированные элементы	47
Полужирное и курсивное форматирование	48
Подкорректированные элементы.....	49
Стандартизированные элементы	50
Современное использование HTML5.....	51
Поддерживает ли браузер вашу разметку?	52
Статистика популярности браузеров.....	55
Определение возможностей с помощью Modernizr	57
Замена отсутствующих возможностей заполнителями	60
Глава 2. Новый способ структурирования страниц	62
Что такое семантические элементы?.....	63
Модифицирование традиционной HTML-страницы	65
Структурирование страницы старым способом	66
Структурирование страницы с помощью HTML5	69
Подзаголовки, созданные элементом <code><hgroup></code>	72
Вставка рисунков с помощью элемента <code><figure></code>	73
Добавление боковой панели с помощью элемента <code><aside></code>	76
Браузерная совместимость для семантических элементов.....	77
Разработка сайта с использованием семантических элементов.....	80
Верхние колонтитулы	80
Создание навигационных ссылок с помощью элемента <code><nav></code>	83
Нижние колонтитулы.....	88
Блоки	91
Система HTML5 для создания схемы документа	92
Как просмотреть схему веб-страницы?	93
Базовые схемы.....	94
Элементы для создания блоков.....	96
Решение проблемы со схемой.....	98
Глава 3. Разметка со смыслом	102
Повторение семантических элементов	103
Обозначение дат и времени с помощью элемента <code><time></code>	104
JavaScript-вычисления и элемент <code><output></code>	106
Выделение текста цветом с помощью элемента <code><mark></code>	107
Другие стандарты, улучшающие семантику.....	109
Стандарт Accessible Rich Internet Applications.....	110
Стандарт Resource Description Framework	111
Микроформаты.....	111
Обозначение контактной информации с помощью микроформата hCard.....	112
Обозначение событий с помощью микроформата hCalendar.....	117
Микроданные	118

Расширенные фрагменты страницы	121
Расширенные результаты поиска.....	122
Движок для поиска кулинарных рецептов	126

ЧАСТЬ II. СОЗДАНИЕ СОВРЕМЕННЫХ ВЕБ-СТРАНИЦ 131

Глава 4. Продвинутое веб-формы 133

Что такое форма?	134
Модернизация традиционной HTML-формы	136
Добавление подсказок	140
Фокус: правильное начало	142
Проверка: ошибкам — нет!.....	142
Как работает проверка HTML5?	143
Отключение проверки	145
Оформление результатов проверки	146
Проверка с помощью регулярных выражений	147
Специализированная проверка	149
Поддержка проверки браузерами	150
Новые типы элемента <i><input></i>	154
Адреса электронной почты	156
URL-адреса	157
Поля поиска	157
Телефонные номера	158
Числа	158
Ползунки	159
Дата и время	160
Цвет	161
Новые элементы	162
Подсказки ввода <i><datalist></i>	162
Индикатор выполнения <i><progress></i> и счетчик <i><meter></i>	165
Элементы <i><command></i> и <i><menu></i> для создания кнопок команд и меню	168
Веб-страница как HTML-редактор	168
Редактирование элементов с помощью атрибута <i>contentEditable</i>	169
Редактирование страницы с помощью атрибута <i>designMode</i>	171

Глава 5. Аудио и видео 174

Основные сведения о воспроизведении видео в современных программах.....	175
Представляем видео и аудио HTML5.....	176
Воспроизведение аудио с помощью элемента <i><audio></i>	177
Воспроизведения видео с помощью элемента <i><video></i>	179
Войны форматов и резервные решения	181
Знакомимся с форматами	182
Поддержка браузерами форматов мультимедиа	184
Множество форматов: как понравиться всем браузерам.....	186
Элемент <i><source></i>	187
Резервное решение Flash	188
Управление плеером посредством JavaScript.....	193
Добавление звуковых эффектов	193
Создание своего видеопроигрывателя	197

Проигрыватели на JavaScript.....	201
Субтитры и доступность.....	203
Глава 6. Основы рисования на холсте.....	205
Базовые возможности холста.....	206
Прямые линии.....	208
Пути и фигуры.....	211
Кривые линии.....	214
Трансформации.....	217
Прозрачность.....	220
Создание простой программы рисования.....	223
Подготовка к рисованию.....	223
Рисование на холсте.....	225
Сохранение содержимого холста.....	227
Совместимость холста с браузерами.....	231
Холст на заполнителе.....	231
Резервное решение для холста и определение возможностей.....	233
Глава 7. Продвинутое методы работы с холстом.....	235
Что еще можно рисовать на холсте?.....	235
Вставка в холст изображений.....	236
Обрезка, разрезка и изменение размеров изображения.....	238
Вставка в холст текста.....	240
Тени и вычурные заливки.....	241
Создание теней.....	242
Заполнение фигур изображениями.....	244
Градиентная заливка фигур.....	245
Обобщая сказанное: рисуем график.....	249
Как сделать фигуры интерактивными?.....	254
Отслеживание нарисованного содержимого.....	255
Проверка на столкновение посредством сравнения координат.....	259
Анимация на холсте.....	261
Простая анимация.....	262
Анимация нескольких объектов.....	263
Практический пример: игра "Лабиринт".....	268
Подготовительные работы.....	270
Анимация значка.....	272
Проверка на столкновение с использованием цвета пикселей.....	274
Глава 8. Совершенствование стилей с помощью CSS3.....	278
Современное использование CSS3.....	279
Стратегия 1: используйте то, что можно.....	279
Стратегия 2: рассматривайте возможности CSS3 как усовершенствования.....	280
Стратегия 3: добавляйте резервные решения с помощью Modernizr.....	281
Стили, специфичные для конкретных браузеров.....	284
Типография для Интернета.....	286
Форматы веб-шрифтов.....	287
Наборы шрифтов.....	289
Веб-шрифты Google.....	292

Использование своих шрифтов.....	295
Размещение текста в несколько колонок	296
Адаптация к разным устройствам	298
Запросы о возможностях отображения	299
Продвинутые запросы о возможностях	303
Полная замена таблицы стилей.....	305
Распознавание мобильных устройств.....	306
Рисование эффектных рамок	308
Прозрачность.....	308
Скругление углов	310
Фон	311
Тени.....	313
Градиенты	314
Создание эффектов перехода.....	316
Простой цветовой переход.....	317
Еще несколько идей с переходами	319
Трансформации	320

ЧАСТЬ III. СОЗДАНИЕ ИНТЕЛЛЕКТУАЛЬНЫХ ВЕБ-ПРИЛОЖЕНИЙ..... 325

Глава 9. Хранение данных	327
Основы веб-хранилища	328
Сохранение данных.....	329
Практический пример: сохранение текущего состояния игры	331
Поддержка веб-хранилища браузерами	334
Продвинутые методы работы с веб-хранилищем	334
Удаление элементов.....	334
Поиск всех сохраненных элементов.....	335
Сохранение чисел и дат	336
Сохранение объектов.....	337
Реагирование на изменения в хранилище.....	339
Чтение файлов	341
Получение файла.....	342
Поддержка браузерами интерфейса File API.....	342
Чтение текстового файла.....	343
Замена элемента <code><input></code>	345
Одновременное считывание нескольких файлов	346
Чтение файла изображения	346
Глава 10. Автономные приложения.....	351
Кэширование файлов с помощью манифеста.....	352
Создание манифеста	353
Использование манифеста.....	356
Помещение манифеста на веб-сервер	356
Обновление файла манифеста.....	359
Браузерная поддержка автономных приложений.....	362
Практические методы кэширования	362
Доступ к онлайн-файлам	363
Добавление резервных решений.....	364

Проверка подключения	366
Информирование об обновлениях с помощью JavaScript	368
Глава 11. Взаимодействие с веб-сервером	371
Отправка сообщений на веб-сервер	372
Объект <i>XMLHttpRequest</i>	373
Отправка запроса веб-серверу	374
Создание сценария	374
Обращение к веб-серверу	376
Получение нового содержимого	378
Отправляемые сервером события	382
Формат сообщений	383
Отправка сообщений с помощью серверного сценария	384
Обработка сообщений в веб-странице	387
Опрос посредством серверных событий	388
Веб-сокеты	390
Получение доступа к веб-сокетам	391
Простой клиент веб-сокетов	393
Примеры веб-сокетов в сети	394
Глава 12. Несколько полезных возможностей на JavaScript	396
Геолокация	397
Принцип работы геолокации	398
Определение координат посетителя	401
Обработка ошибок	403
Установка параметров геолокации	405
Отображение карты	406
Отслеживание перемещений посетителя	410
Фоновые вычисления	411
Трудоемкая задача	413
Выполнение вычислений в фоновом режиме	415
Обработка ошибок веб-работников	419
Отмена исполнения фоновой задачи	419
Обмен более сложными сообщениями	420
Управление историей просмотров	423
Проблем с URL	424
Традиционное решение: hashbang URL	425
HTML5-решение: история сеансов	426
Поддержка браузерами истории сеансов	429
ЧАСТЬ IV. ПРИЛОЖЕНИЯ	433
Приложение 1. Очень краткое введение в CSS	435
Добавление стилей в веб-страницу	435
Анатомия таблицы стилей	436
Свойства CSS	437
Форматирование элементов посредством классов	438
Комментарии в таблицах стилей	439

Продвинутые таблицы стилей.....	439
Структурирование страницы с помощью элементов <code><div></code>	439
Множественные селекторы	440
Контекстные селекторы.....	441
Идентификаторы	442
Селекторы псевдоклассов	443
Селекторы атрибутов	444
Экскурсия по таблице стилей.....	445
Приложение 2. Очень краткое введение в JavaScript	450
Принципы работы JavaScript в веб-странице	451
Вставка кода в разметку	451
Использование функций	453
Перемещение кода JavaScript в файл сценариев	455
Реагирование на события	456
Несколько основных структур языка JavaScript.....	458
Переменные	458
Значение <i>null</i>	459
Область видимости переменных.....	459
Типы данных переменных.....	460
Арифметические операции.....	461
Условные переходы	462
Циклы	464
Массивы	465
Функции, которые получают и возвращают данные.....	466
Взаимодействие со страницей	467
Манипулирование элементами	468
Динамическое подключение к событию	470
Подставляемые в строку функции	472
Предметный указатель	475

Об авторе



Мэтью Мак-Дональд (Matthew MacDonald) — автор научных и технических книг, из-под пера которого вышло свыше десятка книг. Начинающие веб-разработчики могут войти в интернет-сообщество с помощью его книги "Создание веб-сайта. Основное руководство"¹ (Creating a Website: The Missing Manual). Офисные тудяги могут щелкать числа, как орешки, с книгой "Excel 2007. Недостающее руководство"² (Excel 2010: The Missing Manual).

А люди разумные всех сфер деятельности могут сделать для себя открытие, насколько странными они являются, с помощью книг "Ваш мозг. Недостающее руководство" (Your Brain: The Missing Manual) и "Ваше тело. Недостающее руководство" (Your Body: The Missing Manual).

Благодарности

Никакому автору не было бы под силу написать книгу без небольшой армии помощников. Я в неоплатном долгу перед всей командой проекта "Missing Manual", особенно перед моим редактором Нэн Барбер, которая никогда не терялась в ползучих песках HTML5, и перед профессиональными техническими рецензентами Шелли Пауэрс и Стивом Сюэрингом, которые помогали выявлять вкравшиеся ошибки и давали неизменно хорошие советы. И, как всегда, я признателен многим помощникам, которые усиленно работали, индексируя страницы, создавая рисунки и вычитывая окончательный вариант текста.

Наконец, за моменты моей жизни, проведенные вне этой книги, я бы хотел сказать спасибо всем членам моей семьи. Это мои родители, Нора и Поль, мои крёстные родители, Разя и Хамид (Razia и Hamid), моя жена Фариа (Faria) и мои дочери Мая и Бренна (Maya и Brenna). Спасибо вам всем!

Мэтью Мак-Дональд

¹ Мак-Дональд М. Создание Web-сайтов. Основное руководство. — М.: Эксмо, 2010.

² Мак-Дональд М. Excel 2007. Недостающее руководство. — СПб.: БХВ-Петербург, Русская редакция, 2008.

Введение

С первого взгляда можно предположить, что HTML5 — это пятая версия языка HTML для создания веб-страниц. Но в действительности все не так просто.

HTML5 — неформал. Его придумала группа вольнодумцев, которые не входили в группу, отвечавшую за официальный стандарт HTML. В стандарте HTML5 разрешаются методы написания страниц, которые были запрещены десять лет тому назад. В нем подробно изложены инструкции браузерам, как обрабатывать ошибки в разметке страниц, чтобы попытаться отобразить эти страницы, вместо того чтобы сразу же забраковать их. Он, наконец, позволяет воспроизведение видео, не прибегая к помощи модулей расширения браузера, таких как, например, Flash. Также в этом стандарте вводится лавина функциональностей, движимых JavaScript, которые могут придать веб-страницам определенные расширенные, интерактивные возможности, встречаемые в программном обеспечении для настольных компьютеров.

Разобраться в HTML5 — задача не из легких. Самой большой проблемой является то обстоятельство, что термин "HTML5" используется для обозначения свыше дюжины отдельных стандартов. (Как мы узнаем, эта ситуация является результатом эволюции HTML5, который начинался как единый стандарт, но впоследствии был разделен на более управляемые части.) В действительности, термин "HTML5" сейчас означает "HTML5 и связанные стандарты" и, в еще более широком понимании, "следующее поколение технологий разработки веб-страниц". Вот такую версию HTML5 мы и будем рассматривать в этой книге: все от базового языка HTML5 до новых возможностей, сброшенных в одну кучу с HTML5, хотя они *никогда* и не были частью этого стандарта.

Это приводит нас ко второй потенциальной проблеме с HTML5 — браузерной поддержке. Разные браузеры поддерживают различные части HTML5; кроме этого, есть такие возможности, которые не работают ни в одном из браузеров.

Несмотря на все эти трудности, никто не ставит под сомнение, что HTML5 — это будущее веб-дизайна. Он поддерживается крупными компаниями — разработчиками программного обеспечения, такими как Apple и Google; организация W3C (World Wide Web Consortium, Консорциум Всемирной паутины) прекратила свою работу над стандартом XHTML, чтобы формализовать и поддерживать стандарт HTML5; кроме этого, все разработчики браузеров поддерживают значительную

часть этого стандарта. Прочитав эту книгу и усвоив ее материал, вы также сможете стать членом клуба веб-новаторов и создавать искусные веб-страницы, подобные показанной на рис. В1.

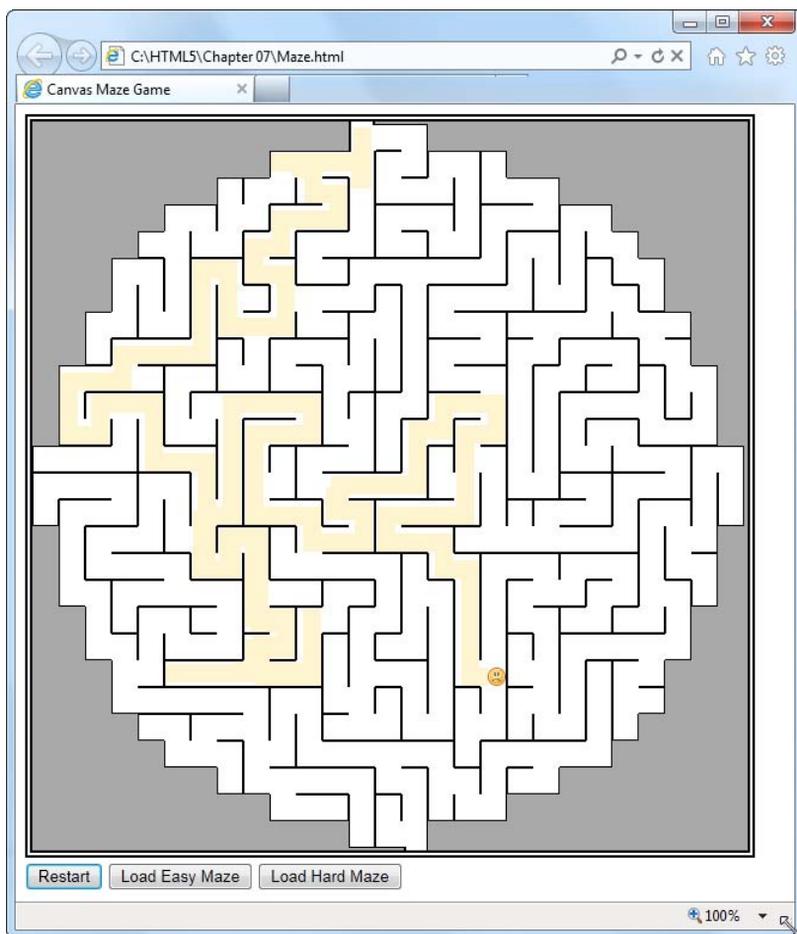


Рис. В1. В темные времена Интернета (иными словами, где-то около года тому назад) для создания игр на веб-страницах нужно было обращаться к помощи модулей расширения браузера, таких как, например, Flash. Но благодаря новым возможностям HTML5, включая холст (применение которого показано на этом рисунке), теперь это можно делать с помощью надежного языка JavaScript без модулей расширения. На этом рисунке HTML5 приводит в действие игру "Лабиринт", которая рассматривается в *главе 7*

Что нужно знать для работы с этой книгой?

В этой книге рассматривается HTML5 — самая последняя и самая мощная версия стандарта HTML. И хотя вам не обязательно быть профессионалом по разметке, нужно все-таки иметь определенные навыки в области веб-разработки. Далее приведено описание основных необходимых навыков.

- **Создание веб-страниц.** Предполагается, что у вас есть опыт создания хотя бы нескольких веб-страниц или, по крайней мере, вы понимаете, как использовать элементы HTML для структурирования содержимого в заголовки, абзацы и списки. Если вы полностью несведущи в области веб-разработки, вам лучше начать с чего-то более легкого.
- **Опыт работы с таблицами стилей.** Современный веб-сайт невозможен без применения технологии CSS (Cascading Style Sheet, каскадные таблицы стилей), посредством которой веб-страницы компонуются и форматируются. Чтобы понимать излагаемый в этой книге материал, вам нужно знать основы таблиц стилей — как создавать их, что помещается внутрь их, а также как подключать их к веб-странице. Если ваши знания по этой теме несколько туманны, можете почерпнуть основные сведения в *приложении 1*. Но если материала из *приложения 1* вам недостаточно или же вы хотите повысить уровень своих знаний в этой области до степени, позволяющей создавать по настоящему искусные компоновки и стили, воспользуйтесь какой-либо дополнительной книгой.
- **Опыт работы с JavaScript.** Нет, для создания веб-страниц язык JavaScript не требуется. Но если вы хотите использовать многие ловкие возможности HTML5, такие как рисование на холсте или взаимодействие с веб-сервером, без JavaScript вам не обойтись. Если у вас имеется хоть какой-либо опыт программирования, но вы не знакомы с JavaScript, тогда материал в *приложении 2* поможет вам чуть-чуть разбираться в этом языке. Но если ваши познания программирования нулевые, тогда вы не сможете должным образом понять многие рассматриваемые в этой книге примеры. В таком случае вам следует ознакомиться с более подробным введением в программирование вообще и в программирование на JavaScript в частности.

Эти требования, к сожалению, не обойти стороной. Такова стоимость возможности работать с передовыми технологиями веб-дизайна.

Написание кода HTML5

Разметку HTML5 можно создавать с помощью тех же приложений, что и разметку обычного HTML. Эти приложения могут быть такими простыми, как базовый текстовый редактор Блокнот (для платформ Windows) или TextEdit (для платформ Mac). Многие современные средства разработки (такие как Adobe Dreamweaver или Microsoft Expression Web) имеют шаблоны, посредством которых можно быстро создавать новые документы HTML5. Но базовая структура страницы HTML5 настолько простая, что ее можно создать с помощью любого веб-редактора, даже если этот веб-редактор не предназначен для работы именно с HTML5.

ПРИМЕЧАНИЕ

И, конечно же, не имеет никакого значения, на какой платформе создавать и просматривать веб-страницы, будь то платформа Windows или самая последняя версия Mac OS X — HTML5 поддерживает операционные системы всех типов.

Просмотр страницы HTML5

В отношении HTML5 часто задается вопрос: какие браузеры поддерживают этот стандарт? К сожалению, на этот вопрос нет четкого ответа. Как мы увидим в этой книге, HTML5 в действительности является коллекцией независимых стандартов. Некоторые его части уже поддерживаются, а некоторые не будут поддерживаться еще в течение нескольких лет (а какие-то, скорее всего, вообще не будут поддерживаться никогда). Ситуация с остальными частями пока не ясна. Это означает, что HTML5 работает на некоторых версиях некоторых браузеров.

Вот список основных браузеров, которые поддерживают значительную часть возможностей HTML5, не требуя резервных решений:

- Internet Explorer 9 и более поздние версии;
- Firefox 3.5 и более поздние версии;
- Google Chrome 8 и более поздние версии;
- Safari 4 и более поздние версии;
- Opera 10.5 и более поздние версии.

Уровень поддержки браузерами HTML5 возрастает с увеличением версии браузера. Например, уровень поддержки HTML5 в Firefox 5 выше, чем в Firefox 3.5.

До рекомендации применять новую возможность HTML5 в книге ясно указывается текущий уровень поддержки этой возможности основными браузерами. Но разработчики браузеров выпускают новые версии сравнительно часто, поэтому вам следует проводить собственные исследования по текущей поддержке, прежде чем применять в своей странице возможность, которая способна вызвать проблемы. В этом отношении можно воспользоваться услугами веб-сайта <http://caniuse.com>, который предоставляет точную информацию о поддержке определенных возможностей конкретными браузерами. (Это полезное средство рассматривается более подробно в разд. "Поддерживает ли браузер вашу разметку?" главы 1.)

ПРИМЕЧАНИЕ

В этой книге рассматриваются некоторые возможности, о которых заведомо известно, что они не работают в некоторых браузерах. Не впадайте по этому поводу в панику. Это вполне приемлемо, если вы хотите только получить представление обо всем диапазоне возможностей HTML5, но при этом фокусироваться на тех из них, которые можно использовать уже сейчас. А "сырые" функциональности можно рассматривать как демонстрацию того, что нас ожидает в Интернете в будущем.

Когда HTML5 будет готов?

Общий ответ на этот вопрос таков: он уже готов сейчас. Даже постылый Internet Explorer 6, которому 10 лет и который битком набит своеобразными функциями обработки веб-страниц, часто должным образом не отображающими эти страницы, может отображать элементы HTML5. Это достигается тем, что стандарт HTML5 преднамеренно разрабатывался так, чтобы включать и расширять традиционный HTML.

Как мы уже узнали, HTML5 в действительности является коллекцией разных стандартов с разным уровнем поддержки браузерами. Поэтому, хотя любой веб-разработчик может переключиться на использование документов HTML5 уже сегодня (и некоторые крупные сайты, такие как Google, YouTube и Wikipedia, уже перешли), может пройти несколько лет, прежде чем использование большей части новых хитроумных возможностей HTML5 станет безопасным. По крайней мере, без добавления какого-либо механизма резервных решений для менее сообразительных браузеров.

ПРИМЕЧАНИЕ

По большому счету не имеет особого значения, к какой спецификации принадлежит та или иная возможность. Важна современная поддержка браузерами этой возможности, а также вероятность ее поддержки в будущем. Когда в этой книге рассматривается какая-либо новая возможность, указывается, в какой спецификации она определяется, и текущий уровень ее поддержки основными браузерами.

Разработчикам, которые трепетно относятся к стандартам, может быть интересно знать, насколько близки различные стандарты к моменту объявления официального статуса. Дать точный ответ на этот вопрос сложно, поскольку создатели HTML5 считают более важным то, что поддерживается браузерами, нежели то, что указано в стандарте. Иными словами, веб-разработчики могут использовать по желанию любые возможности уже сейчас, если могут заставить их работать. Но веб-разработчики, крупные компании, государственные органы и другие организации часто составляют свое мнение о готовности языка к применению по официальному статусу его стандарта.

Вообще, в настоящее время спецификация HTML5 находится в руках организации W3C в качестве *рабочего проекта* (working draft). Данное определение обозначает, что это вполне установившийся стандарт, но такой, что еще может измениться в процессе прохождения этапа *возможной рекомендации* (candidate recommendation), в который он войдет где-то в 2012 г. А вхождение в стадию *рекомендации* (recommendation) может занять многие годы, т. к. для этого стандарту требуется пройти многочисленные тестирования. Но это уже не так важно, т. к. на этом этапе изменения будут немногочисленные, и все, кто хочет использовать HTML5, уже сядут на этот поезд.

О чем эта книга?

Двенадцать глав этой книги содержат всеохватывающее руководство по HTML5.

Часть I. Знакомство с новым языком.

- В главе 1 "*Представляем HTML5*" рассматривается, как HTML превратился в HTML5. Мы познакомимся с первым документом HTML5, увидим, какие изменения претерпел язык, а также обсудим поддержку возможностей браузерами.
- В главе 2 "*Новый способ структурирования страниц*" рассматриваются семантические элементы HTML5 — группа элементов, которые могут придать смысл

разметке. При правильном использовании эта дополнительная информация может оказаться полезной для браузеров, средств чтения экрана, инструментов для веб-дизайна и поисковых систем.

- В главе 3 "*Разметка со смыслом*" обсуждается семантика и рассматриваются дополнительные стандарты, такие как *микроданные*. И хотя эти сведения могут показаться несколько теоретизированными, веб-разработчиков, которые разберутся в них, ожидает достойное вознаграждение — лучшие, более подробные результаты поисковых запросов в поисковых системах, таких как Google.

Часть II. Создание современных веб-страниц.

- В главе 4 "*Продвинутые веб-формы*" исследуются изменения, внесенные HTML5 в элементы веб-форм — текстовые поля, списки, флажки и прочие элементы управления, используемые для сбора информации от посетителей страницы. HTML5 добавляет несколько новых примочек и некоторые базовые средства для улавливания ошибок ввода.
- В главе 5 "*Аудио и видео*" изучается одна из самых захватывающих возможностей HTML5 — поддержка воспроизведения видео и аудио. Мы научимся выживать в битве видеокodeков для веба, чтобы создавать страницы, воспроизводящие мультимедиа во всех браузерах, и даже увидим, как создать собственный проигрыватель.
- В главе 6 "*Основы рисования на холсте*" рассматривается двумерная поверхность для рисования — холст. На этом холсте мы научимся рисовать фигуры, изображения и текст, и даже на его основе создадим простую программу рисования (применив JavaScript).
- В главе 7 "*Продвинутые методы работы с холстом*" мы разовьем наши навыки работы с холстом и узнаем, как создавать тени и вычурные узоры, а также освоим более продвинутые методы, такие как создание анимации и фигур, активируемых щелчком мыши.
- В главе 8 "*Совершенствование стилей с помощью CSS3*" рассказывается о последней версии стандарта CSS, которая хорошо дополняет HTML5. Мы научимся украшать текст вычурными шрифтами, настраивать страницу для отображения на разных типах мобильных устройств и добавлять бросающиеся в глаза эффекты с помощью переходов.

Часть III. Создание интеллектуальных веб-приложений.

- В главе 9 "*Хранение данных*" рассматривается новая функциональность — веб-хранилище, которое позволяет странице сохранять информацию на компьютере посетителя. (Оно похоже на cookies, но на порядок эффективнее.) Мы также научимся обрабатывать выбранные пользователем файлы посредством JavaScript-кода прямо в веб-странице, а не отправляя их для этого на сервер.
- В главе 10 "*Автономные приложения*" исследуется новая функциональность кэширования, которая позволяет браузеру работать с веб-страницей даже при отсутствии подключения к Интернету.

- В главе 11 "Взаимодействие с веб-сервером" мы познакомимся со сложным обменом информацией с веб-сервером. Мы начнем с изучения испытанного временем объекта XMLHttpRequest, который позволяет посредством JavaScript отправлять веб-серверу запросы информации. После этого мы перейдем к рассмотрению двух новых возможностей: отправляемых сервером событий и более амбициозной, но не совсем доведенной, функциональности веб-сокетов.
- В главе 12 "Несколько полезных возможностей на JavaScript" рассматриваются три дополнительные возможности, направленные на решение задач современного Интернета. Сначала мы увидим, как использовать геолокацию, чтобы установить местоположения посетителя страницы. Потом мы воспользуемся веб-работниками для выполнения трудоемких задач в фоновом режиме. Наконец, мы исследуем функциональность отслеживания истории просмотров страниц, которая позволяет синхронизировать URL динамически обновляемой страницы с ее текущим содержанием.

Кроме этого, два приложения помогут вам получить базовые знания, необходимые для овладения HTML5. В *приложении 1* дается краткий обзор CSS, а в *приложении 2* — JavaScript.

Онлайновые ресурсы

Купив эту книгу, вы приобрели больше, чем просто книгу для чтения. Вместе с ней вы получили доступ к онлайн-файлам примеров, а также советам, статьям и паре-другой видео.

Недостающий CD

Эта книга не имеет сопровождающего компакт-диска, но это не значит, что вы что-либо теряете. Все рассматриваемые в этой книге примеры можно загрузить по адресу <http://missingmanuals.com/cds/html5mm>. Кроме этого, чтобы вам не стирать пальцы, вводя URL упоминаемых в книге веб-сайтов, все они предоставлены в виде ссылок на указанной веб-странице.

СОВЕТ

Если вы ищете конкретный пример, самый быстрый способ найти его — посмотреть на соответствующий рисунок в книге. В конце строки адреса обычно будет название файла. Например, на рис. 1.1 указывается путь c:\HTML5\Chapter01\SuperSimpleHTML5.html, что говорит нам, что файл примера называется SuperSimpleHTML5.html.

Сайт для тестирования примеров книги

Рассматриваемые в книге примеры можно испытать еще одним способом — на странице по адресу <http://www.prosetech.com/html5/>. Эта страница содержит все примеры из данной книги, которые можно открыть в браузере. Просмотр примеров с этой страницы может помочь вам избежать многих проблем, т. к. HTML5 содер-

жит массу возможностей, для работы которых страница должна предоставляться веб-сервером. (Если попытаться открыть страницу с такими возможностями с локального жесткого диска, они могут отработать с непредвиденным результатом или же не работать вообще.) Просматривая пример с указанного веб-сайта, вы сможете увидеть, как он должен работать, перед тем, как загрузить страницу себе на компьютер и начать экспериментировать с ней.

ПРИМЕЧАНИЕ

Не волнуйтесь о примерах, которые нужно запускать с веб-сервера — в книге предупреждается о таких примерах.

ЧАСТЬ I

Знакомство с новым языком

Глава 1. Представляем HTML5

Глава 2. Новый способ структурирования страниц

Глава 3. Разметка со смыслом

ГЛАВА 1

Представляем HTML5

Историю развития языка HTML можно сравнить с детективным рассказом в том, что в ней есть свой неожиданный поворот событий, в результате которых появилась новая версия языка — HTML5.

Предполагалось, что язык HTML уйдет в небытие, не дожив до XXI столетия. Организация W3C (World Wide Web Consortium, Консорциум Всемирной паутины), которая занимается разработкой и внедрением официальных стандартов Всемирной паутины, забросила язык HTML в далеком 1998 г., считая его не способным на дальнейшее выживание. Свои надежды на будущее консорциум W3C возлагал на модернизированного наследника HTML — язык XHTML. Но язык HTML не умер. Его "подобрала" группа программистов-аутсайдеров и не только возвратила его к жизни, но и заложила основу для новых возможностей, которые мы с вами и исследуем в этой книге.

В данной главе мы выясним, почему язык HTML был брошен умирать от старости и как он был возвращен к жизни. Мы узнаем основные принципы и возможности языка HTML5, а также рассмотрим тернистую проблему поддержки этого языка разными браузерами. Кроме этого, мы также рассмотрим настоящий документ HTML5, как его самую простую форму, так и более практический шаблон, который можно использовать в качестве отправной точки для создания любого веб-сайта.

История языка HTML5

Как вы знаете, HTML — это язык для создания веб-страниц. Ключевая идея языка HTML — организация содержимого с помощью *элементов* — не претерпела никаких изменений с самых ранних времен Всемирной паутины. Более того, даже очень старые веб-страницы без проблем обрабатываются в наиболее современных браузерах (включая и те, которые не существовали на момент создания этих страниц, например Firefox или Chrome).

Но успех и почтенный возраст также несут с собой определенные существенные угрозы. Что и случилось с языком HTML — в 1988 г. консорциум W3C прекратил

его поддержку и попытался заменить его языком на основе языка XML — XHTML 1.0.

Язык XHTML 1.0: строго по правилам

В стандарте XHTML используются те же синтаксические соглашения, что и в HTML, но в нем ужесточены требования к следованию установленным правилам. Большая часть отступлений от правил разметки, которые сходят с рук в традиционном HTML, попросту неприемлемы в XHTML.

Например, допустим, что вы хотите выделить курсивом последнее слово заголовка следующим образом:

```
<h1>Из жизни <i>уток</i></h1>
```

Но при этом вы случайно поменяли местами два последних тега:

```
<h1>Из жизни <i>уток</h1></i>
```

Когда браузер сталкивается с этой слегка подпорченной разметкой, он в состоянии "понять", что вы действительно имели в виду, и без малейших претензий выделяет последнее слово курсивом. Но несовпадающие теги нарушают официальные правила XHTML. Если проверить эту страницу в валидаторе формата XHTML (или открыть ее в какой-либо программе для разработки веб-сайтов, например Dreamweaver), то будет выведено сообщение, указывающее на эту ошибку. В отношении разработки веб-сайтов это полезно, т. к. позволяет отловить небольшие ошибки, вследствие которых страница может отображаться по-разному в различных браузерах, или которые могут вызвать более серьезные проблемы при попытке отредактировать страницу с целью ее улучшения.

Поначалу XHTML пользовался большим успехом. Профессиональные веб-разработчики, раздосадованные индивидуальными особенностями отображения страниц браузерами и вседозволенностью в области веб-разработки, скопом переходили на XHTML. При этом они были вынуждены совершенствовать свои навыки работы и отказываться от использования значительного количества полусырых возможностей форматирования, предоставляемых HTML. Но многие из предполагаемых преимуществ XHTML — такие как функциональная совместимость с инструментами XML, облегчение обработки страниц автоматизированными программами, переносимость на мобильные платформы, а также расширяемость самого языка XHTML — так никогда и не материализовались.

Тем не менее XHTML стал стандартом для большинства серьезных веб-разработчиков. Но в то время, как все казались достаточно довольными этим языком разметки, у него был один скелет в шкафу: хотя браузеры понимали разметку XHTML, они не обеспечивали строгую проверку отступлений от правил, требуемую этим стандартом. Это означает, что страница может быть создана с нарушением правил XHTML, но браузеры и глазом не моргнут при ее обработке. Более того, ничто не могло помешать веб-разработчику бросить в одну кучу небрежно написанную разметку и устаревшее HTML-содержимое и назвать все это страницей XHTML. Ни один браузер на планете не имел бы никаких претензий при обработке

такой страницы. И это обстоятельство вызывало у людей, ответственных за стандарт XHTML, глубокое чувство тревоги.

XHTML 2: неожиданный провал

Эта проблема должна была, по идее, быть решенной в следующей версии — XHTML 2. В ней нужно было ужесточить правила обработки ошибок, которые бы заставляли браузеры не обрабатывать страницы, не отвечающие стандарту XHTML 2. В XHTML 2 также избавились от многих странностей и соглашений, унаследованных от HTML. Например, система нумерации заголовков (`<h1>`, `<h2>`, `<h3>` и т. д.) была заменена одним элементом `<h>` с уровнем обозначаемого им заголовка, зависящего от местонахождения этого элемента в веб-странице. Подобным образом элемент `<a>` был заменен возможностью, позволяющей веб-разработчикам преобразовывать любой элемент в ссылку, а вместо атрибута `alt` элемента `` был предложен новый способ предоставления альтернативного содержимого.

Изменения такого рода были типичными для XHTML 2. С теоретической точки зрения, они делали язык более аккуратным и были более понятными. Но с практической стороны, они вынуждали всех изменить свой подход к созданию веб-страниц (не говоря об обновлении уже созданных веб-страниц), не добавляя при этом никакой новой функциональности, чтобы оправдать всю эту работу. Попутно из XHTML 2 было удалено несколько удобных элементов, которые все еще пользовались популярностью среди веб-разработчиков, таких как `` (полужирный текст), `<i>` (текст курсивом) и `<iframe>` (для вложения одной веб-страницы в другую).

Но, возможно, худшей проблемой оказалась чрезвычайно медленная скорость внесения изменений. Разработка XHTML 2 тащилась в течение пяти лет, и интерес разработчиков к этому стандарту медленно угасал.

HTML5: возвращение к жизни

Приблизительно в то же самое время (начиная с 2004 г.) группа разработчиков начала рассматривать будущее Всемирной паутины в другом ракурсе. Вместо того чтобы попытаться разобраться, что было неправильным (или просто "грязным" с философической точки зрения) в HTML, они сфокусировались на том, чего в нем не хватало, что хотели бы иметь веб-разработчики для воплощения своих идей.

В конце концов, HTML зародился как инструмент для отображения документов. С добавлением языка сценариев JavaScript HTML преобразовался в систему для разработки веб-приложений, таких как поисковые движки, онлайн-магазины, картографические инструменты, средства чтения электронной почты и многие другие. Но в то время как искусное веб-приложение может делать много чего впечатляющего, создать такое приложение — задача не из легких. Большинство разработчиков использует для этого мешанину кода JavaScript, один или несколько популярных инструментариев JavaScript, а также веб-приложение, исполняемое на веб-сервере. Добиться правильного и единообразного взаимодействия всех этих

составляющих на разных браузерах — сложная задача. Даже когда все наконец работает, нужно постоянно "присматривать за скотчем и скобами", которые удерживают всю конструкцию.

Такая ситуация вызывала особенную озабоченность среди разработчиков браузеров, поэтому группа дальновидных разработчиков из компании Opera Software (создатели браузера Opera) и компании Mozilla Foundation (создатели браузера Firefox) начали агитировать за включение в XHTML больше возможностей, ориентированных на разработчиков. Когда их попытки не увенчались успехом, компании Opera, Mozilla и Apple создали группу WHATWG (Web Hypertext Application Technology Working Group, рабочая группа по технологии гипертекстовых веб-приложений) с целью работы над новыми решениями.

Группа не ставила перед собой задачу заменить HTML, ее целью было плавное расширение языка, и причем такое, чтобы расширения были обратно совместимыми. Надо сказать, что самая ранняя версия работы этой группы включала две спецификации расширений — Web Applications 1.0 и Web Forms 2.0. В конечном итоге эти стандарты эволюционировали в HTML5.

ПРИМЕЧАНИЕ

Предполагается, что число 5 в названии HTML5 означает: данный стандарт является продолжением стандарта HTML (последней версией стандарта HTML перед XHTML была версия 4.01). Это, конечно же, не совсем верно, т. к. HTML5 поддерживает все разработки, существовавшие в области создания веб-страниц в течение десяти лет после выпуска HTML 4.01, включая строгий синтаксис в стиле XHTML (если разработчики желают использовать его), а также множество инноваций для JavaScript. Тем не менее это название делает ясным следующее: язык HTML5 *может* поддерживать соглашения XHTML, но *требует* следования правилам HTML.

К 2000 г. все интересное происходило в лагере группы WHATWG. После некоторого периода болезненных размышлений организация W3C решила распустить работающую над XHTML 2 группу и работать вместо этого над формализацией стандарта HTML5. На этом этапе первоначальный стандарт HTML5 был разделен на более управляемые части, и многие из его функциональных возможностей стали отдельными стандартами (*см. врезку. "На профессиональном уровне. Что входит в состав HTML5?"* далее).

СОВЕТ

Ознакомиться с официальной версией стандарта HTML5 организации W3C можно на веб-сайте по адресу: www.w3.org/TR/html5.

НА ПРОФЕССИОНАЛЬНОМ УРОВНЕ **Что входит в состав HTML5?**

Браузеров, "поддерживающих" HTML5, не существует. Вместо этого, каждый браузер поддерживает постепенно расширяющееся подмножество возможностей HTML5. Этот подход является как хорошим, так и плохим. Хорош он потому, что браузеры могут быстро реализовать готовые части стандарта HTML5, пока другие возможности продолжают развиваться. А плох он тем, что заставляет веб-разработчиков беспокоиться о проверке, поддерживает ли конкретный браузер каждую функциональную возможность, которую они хотят использовать. (Трудные и не очень трудные методы такой проверки рассматриваются далее в этой книге.)

Далее приводится список и короткое описание основных категорий функциональных возможностей, охватываемых HTML5.

- **Ядро HTML5.** Эта часть HTML5 составляет официальную версию спецификации организации W3C. Она содержит новые семантические элементы (см. главы 2 и 3), новые и усовершенствованные элементы управления для веб-форм (см. главу 4), поддержку аудио и видео (см. главу 5), а также холст для рисования с помощью JavaScript (см. главы 6 и 7). В эту категорию входит большинство функциональных возможностей, которые наилучшим образом поддерживаются браузерами.
- **Ранние возможности HTML5.** Это возможности, которые были реализованы в первоначальной спецификации HTML5, подготовленной группой WHATWG. Большинство из них — это спецификации для возможностей, требующих JavaScript и поддерживающих развитие веб-приложения. Наиболее важными являются локальное хранение данных (см. главу 9), приложения, работающие в автономном режиме (см. главу 10), и обмен сообщениями (см. главу 11), но также несколько других, которые рассматриваются в этой книге.
- **Возможности, иногда называемые HTML5.** Это возможности следующего поколения, которые часто считаются частью HTML5, хотя они никогда не входили в стандарт HTML5. Эта категория включает спецификацию CSS3 (см. главу 8) и геолокацию (см. главу 12).

Путаница с этими стандартами создается не только ничего не сведущими менеджерами и авторами статей о технологиях. Даже сама организация W3C размывает границы между "настоящим" HTML5 (как определено этим стандартом) и "маркетинговой" версией (которая включает все новшества и кухонную раковину). Например, официальный веб-сайт организации W3C по логотипам (www.w3.org/html/logo) призывает разработчиков генерировать логотипы HTML5, продвигающие стандарты CSS3 и SVG-2, разработка которых велась задолго до появления HTML5.

HTML: живой язык

В результате перехода поддержки HTML сначала от организации W3C к группе WHATWG, а потом обратно, возникла довольно необычная ситуация. Технически организация W3C отвечает за определение, что является официальным HTML5, а что — нет. Но в то же самое время группа WHATWG продолжает свою работу, придумывая будущие возможности HTML. Только теперь она называет его не HTML5, а просто HTML, объясняя это тем, что HTML будет продолжать существовать, как *живой язык*.

Так как HTML является живым языком, то HTML-страница никогда не устареет и не перестанет работать. Для HTML-страниц никогда не потребуется номер версии (даже в блоке указания типа документа `<doctype>`), а веб-разработчикам никогда не понадобится обновлять свою разметку от одной версии языка к другой, чтобы она работала на новых браузерах.

Так как HTML — это живой язык, новые возможности (и новые элементы) можно добавлять к стандарту HTML в любое время. Некоторые разработчики могут решить использовать эти возможности в своих веб-страницах, а некоторые разработчики веб-браузеров — поддерживать их в своих продуктах. Но возможности никогда не будут привязаны к какой-либо конкретной версии стандарта.